

# TEMA 1

SISTEMA INFORMÁTICO. ESTRUCTURA  
FUNCIONAL.

# Introdución á Informática

O termo informática procede da conxunción de dúas palabras: "*información*" e "*automática*". A informática permítenos xestionar a información en forma de datos e instrucións, permítenos controlar sistemas de maquinaria, crear e modificar enormes bases de datos, realizar os efectos especiais dunha película, realizar tarefas rutinarias como a confección de documentos, levar a contabilidade dunha empresa, etc.

Podemos definir a informática como: "***Ciencia que estuda o tratamento racional e automático da información ademais da tecnoloxía para mantela conservada e utilízala de maneira eficiente e económica***".

A principal ferramenta que actualmente nos permite o tratamento automático da información é o ordenador, que podemos definir como "***Máquina composta de elementos físicos, na súa maioría de orixe electrónica, capaz de realizar diferentes operacións sobre a información: lectura, almacenamento, cálculos, comparacións e escritura a gran velocidade e con gran precisión***".

# Historia da informática

En 1937 a universidade de Harvard un equipo de enxeñeiros de IBM comezan a desenvolver un proxecto que en 1944 culmina na primeira computadora **electromecánica**, denominada *calculadora automática de secuencia controlada*, aínda que coñecida popularmente como **MARK I**. Tiña 16,6 metros de longo, 2,6 de alto, pesaba 70 toneladas e estaba constituída por 800.000 pezas móbiles, e o seu cableado era de 800.000 metros. Sumaba dous números en menos dun segundo e multiplicábaos en menos de tres. Podía traballar con operandos de ata 23 cifras.

Foron as esixencias militares, concretamente a necesidade de elaborar táboas para o cálculo de traxectoria de proxectís, as que induciron aos científicos estadounidenses a construír o que é considerado primeiro ordenador electrónico, o **ENIAC** (Electronic Numerical Integrator And Calculator), entrou en funcionamento en 1945, e estaba composto por 18.000 válvulas e un consumo de 160 Kw (o mesmo que 2666 bombillas de 60w) . Era mil veces máis rápido que MARK I. En 1951 constrúese a primeira computadora posta á venda, o **UNIVAC I**, a partir de aí disparouse o desenvolvemento de novas máquinas e a súa produción en serie.

# Xeracións de ordenadores

Dende que na primeira parte da década dos cincuenta comezaron a utilizarse os ordenadores con fins comerciais, estes evolucionaron de tal xeito, que cabe destacar cinco xeracións:

**PRIMEIRA XERACIÓN (1940-1952):** Os ordenadores electrónicos da primeira xeración foron deseñados con **VÁLVULAS ELECTRÓNICAS DE BALEIRO**. O seu tamaño era enorme e o seu mantemento moi complicado e a fiabilidade escasa. O tempo medio entre dúas avarías era de media hora. Os tempos de cálculo dos seus circuítos eran de varios microsegundos, polo que a execución de programas longos era de varios días. Utilizaban como linguaxe de programación a **LINGUAXE MÁQUINA**. Como único soporte para almacenar a información utilizaban as **TARXETAS PERFORADAS** ou **CINTA PERFORADA**.

# Xeracións de ordenadores

**SEGUNDA XERACIÓN (1952-1964):** As válvulas son substituídas por **TRANSISTORES**. Tal innovación reduciu considerablemente o tamaño dos ordenadores e aumentou a súa fiabilidade, en consecuencia viuse aumentada a súa capacidade de cálculo e reducido o seu consumo. O seu campo de utilización, ata agora científico e militar, ampliouse o administrativo e universitario.

Ofrecen a posibilidade de simultanear o cálculo puro con operacións de entrada/saída. Aparece por primeira vez a **MEMORIA INTERNA** a base de núcleos de ferrita e como memoria externa os primeiros **SOPORTES MAGNÉTICOS** (a cinta magnética e os tambores magnéticos). Aparecen as **LINGUAXES DE PROGRAMACIÓN** o ensamblador, e algúns de alto nivel (COBOL, FORTRAN, ALGOL.. .)

# Xeracións de ordenadores

**TERCEIRA XERACIÓN (1964-1971):** O elemento máis significativo é o **CIRCUITO INTEGRADO**, que consiste no encapsulamento dunha grande cantidade de compoñentes electrónicos discretos, conformando un ou varios circuítos cunha función determinada, sobre unha pastilla de plástico.

O tamaño dos ordenadores volveu reducirse unha vez máis, aumentando a súa potencia de cálculo e a súa fiabilidade. Pasouse así de ordenadores capaces de executar miles de instrucións por segundo a outros que podían executar millóns delas nese mesmo tempo. A miniaturización estendeuse a todos os circuítos da computadora.

O software evolucionou de forma considerable, coa aparición de varios **SISTEMAS OPERATIVOS**. Comezaron a utilizarse as memorias de semicondutores e os **DISCOS MAGNÉTICOS**.

# Xeracións de ordenadores

**CUARTA XERACIÓN (1971-1981):** En 1971, produciuse unha nova revolución no mundo dos ordenadores, orixinada polo nacemento do **MICROPROCESADOR**. Tecnoloxía de moi alta integración e elevadísima velocidade e fiabilidade, introducindo toda a UCP nunha pastilla.

A tecnoloxía utilizada, permitiu a fabricación de microcomputadores e ordenadores persoais de alta fiabilidade, velocidade e economía. Comezan a utilizarse os **DISQUETES** ou floppy-disk como unidade de almacenamento externo.

Aparecen unha multitude de linguaxes de todo tipo e as **REDES** de transmisión de datos para conectar ordenadores.

# Xeracións de ordenadores

**QUINTA XERACIÓN (1981- ):** Ó aumento constante de velocidade de proceso, e a miniaturización dos seus compoñentes, hai que engadir a proliferación das redes de ordenadores e dispositivos (redes integradas).

Créanse ordenadores con "Intelixencia Artificial". Aparecen as linguaxes naturais (Linguaxes de quinta xeración). Intégranse nun mesmo proxecto datos, imaxes e voz. (MULTIMEDIA). Xeneralízase o uso da rede de redes, INTERNET...



# Xeracións de ordenadores. Resumo

- .En 1945 creouse o primeiro ordenador chamado ENIAC
- .En 1951 ponse a venta o primeiro ordenador chamado UNIVAC I, despois UNIVAC II e UNIVAC III
- .En cada xeracións os ordenadores son máis potentes, pequenos e fiables

Xeración	Ano	Deseñados con:	Empreganse nos campos:	Úsanse	Aparecen	Características
<b>Primeira</b>	1940-1952	Válvulas de baleiro	Científico Militar	Tarxetas perforadas	Linguaxe máquina	Unha soa tarefa, se están facendo unha E/S non fan outra cousa. Tempo medio entre avarías 30 minutos.
<b>Segunda</b>	1952-1964	Transistores	Administración xestión Comercial	Soportes magnéticos secuenciais	S.O. Monousuario e linguaxes de programación	Simultaneaban o cálculo cas operacións de E/S. Procesos por lotes
<b>Terceira</b>	1964-1971	Circuitos integrados	Grandes e Medianas Empresas	Discos magnéticos	S.O. multiusuario	Executan millóns de instrucións por segundo
<b>Cuarta</b>	1971-1981	Circuitos integrados	PYMES	Disquetes	Microprocesador	Aparecen as primeiras Redes
<b>Quinta</b>	1981-hoxe	Circuitos integrados	Todos os campos, e nos fogares	Multimedia, Redes, Internet	Linguaxes naturais e 5º xeración	Ordenadores de intelixencia artificial Linguaxes naturais e MULTIMEDIA

# SISTEMAS INFORMÁTICOS

O conxunto de ordes ou instrucións que se introducen nun ordenador para realizar un proceso determinado denomínase **programa**. O conxunto de varios programas denomínase **aplicación informática**.

Exemplo: Unha aplicación informática pode ser un programa bancario, que consta de varios programas, cada un dos cales ten unha finalidade concreta: nóminas, préstamos, contabilidade...

Pois ben, o conxunto de instrucións, programas e aplicacións informáticas quedan definidos baixo o termo de **software**.

Por outro lado, hai que considerar que para que estes programas funcionen e poidan xerar a información que o usuario precisa, necesítase determinados compoñentes físicos. Estes compoñentes físicos agrúpanse baixo a denominación de **hardware**.

# SISTEMAS INFORMÁTICOS

**Sistema Informático** – Hardware e Software necesario para satisfacer determinadas necesidades do usuario

**Hardware:** son os compoñentes físicos (tanxibles), como por exemplo monitor, teclado, microprocesador, memoria, etc.

**Software:** é o parte intanxible (programas e aplicacións). Pódese dividir en:

- **Software base** - Aquela parte do software sen a cal o computador non pode funcionar. Tamén se denomina **Sistema operativo**.
- **Software de aplicación** - A parte do software que serve para procesar a información.

# SISTEMAS INFORMÁTICOS

**Firmware:** é a parte intanxible dos compoñentes hardware. Por exemplo o software co que están programadas as memorias ROM, o software co que se configuran dispositivos de comunicacións como routers e switches, etc.

Este software non é doadamente modificable, unha vez que se grava nun compoñente hardware queda practicamente invariable ao longo da vida deste.

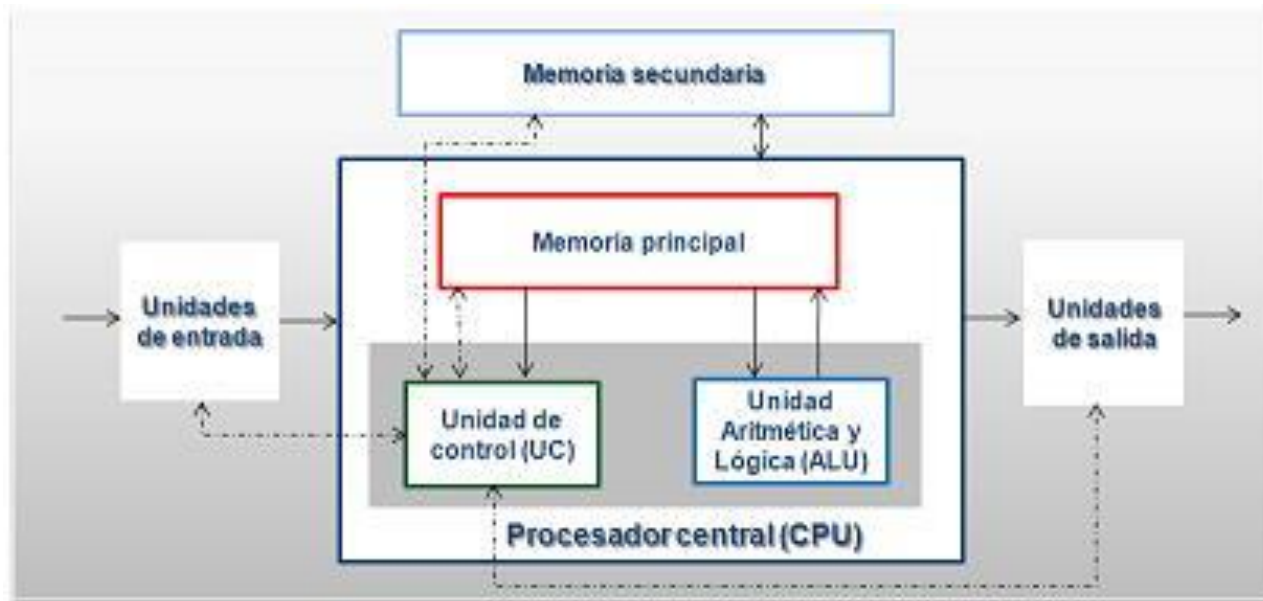
**Persoal informático:** Son as persoas que utilizan, desenvolven ou manteñen o hardware e o software.

# S.I. CAPAS



# ESTRUTURA DO PC. Modelo Von- Neumann

Toda computadora basease no modelo de Von Neumann exposto en 1946 polo matemático húngaro John Louis Von Neumann. Unha computadora está formada por tres partes fundamentais (Microprocesador, Memoria Principal e Unidades de E/S), as cales dende o punto de vista do funcionamento, son independentes.



# ESTRUTURA DO PC. Modelo Von- Newman



Tendese a integrar as cachés nos microprocessadores. A L1 intégrase dende o 89, a L2 dende o 92. Actualmente os microprocessadores incorporan memorias caché L3

# ESTRUTURA DO PC

A estrutura funcional dun ordenador está formada por:

- **Unidade Central de Proceso**(CPU) ou procesador que se encarga de procesar os datos
- **Memoria**, que almacena a información
- **Unidades de entrada e saída**, que permiten o intercambio de datos ou información co exterior, así como o seu almacenamento.

Todos os dispositivos deben estar interconectados para que a información flúa de entre eles segundo sexa necesario; esa é a misión dos **buses**, auténticas canles polas que circula a información.



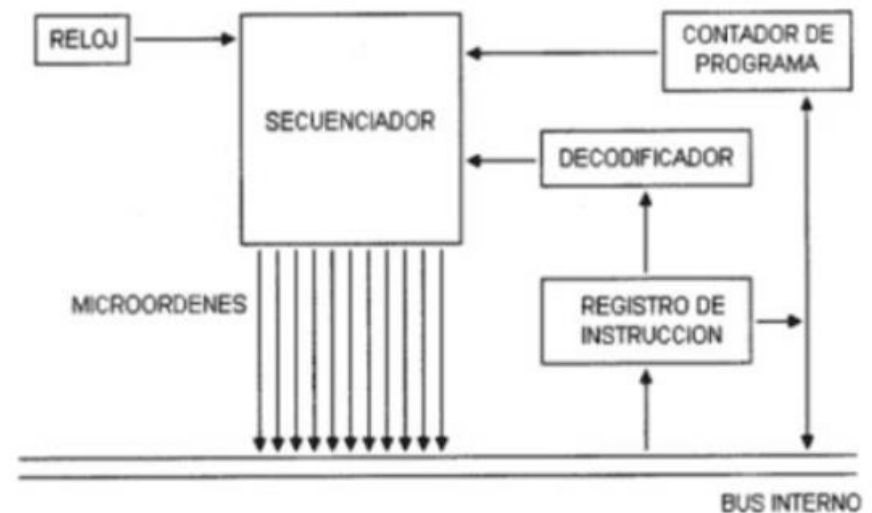
# Unidade de Control - UC

É o centro nervioso do ordenador, xa que dende ela contrólanse e gobiñanse tódalas operacións. A tarefa fundamental é recibir información para interpretala e procesala despois mediante as ordes que envía aos outros compoñentes do ordenador co fin de que cada un actúe no momento e da forma precisa.

Consta dos seguintes elementos:

Unha instrución a hora de ser executada pola computadora da lugar a unha gran cantidade de microordenes

Pasos para a execución dunha instrución



# Unidade de Control - UC

**Contador de programa (CP):** Tamén denominado rexistro de control de secuencia ou contador de Instrución (CI), contén permanentemente a dirección de memoria da seguinte instrución a executar. Ao iniciar a execución dun programa toma a dirección da súa primeira instrución. Incrementa automaticamente o seu valor cada vez que se conclúe unha instrución.

**Rexistro de instrución (RI):** Contén a instrución que se está a executar en cada momento. Esta instrución levará consigo un **código de operación** (dependendo dese código activaranse uns ou outros compoñentes do ordenador) e no seu caso os **operandos** ou as súas direccións de memoria.

**Decodificador:** Encárgase de extraer o código de operación da instrución en curso (que está no RI), analízao e emite os sinais necesarios ao resto de elementos para a súa execución a través do secuenciador.

**Reloxo:** Indica en que momento debe comezar unha determinada operación e en que momento debe finalizar, para iso emite uns impulsos (pulsos de reloxo) a intervalos fixos de tempo que serven como referencia o resto dos compoñentes

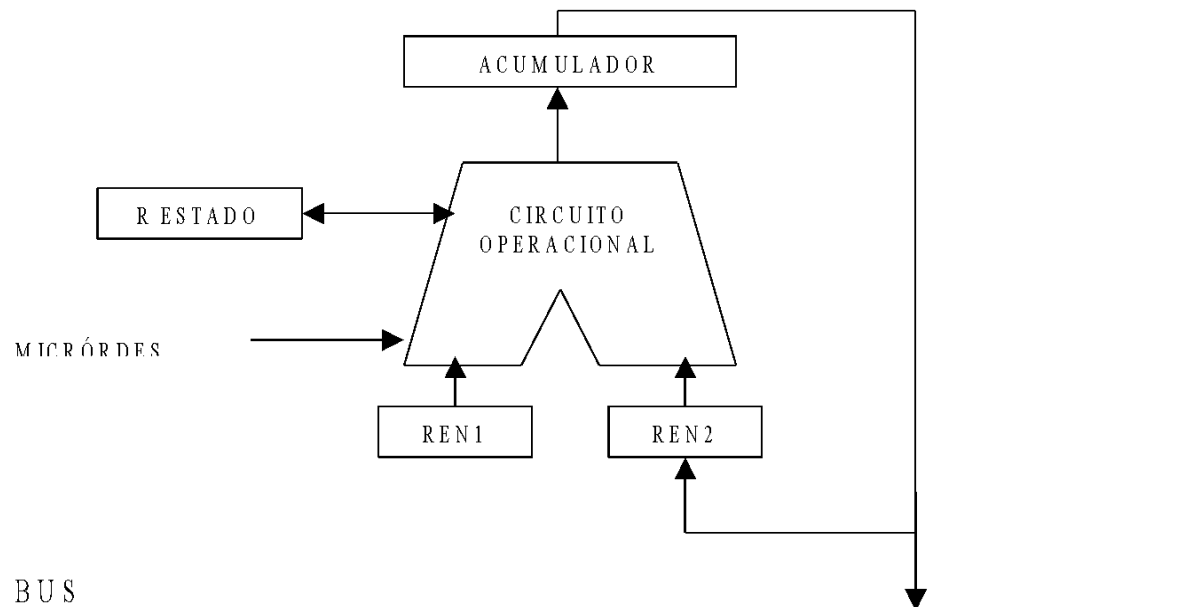
**Secuenciador:** Tamén denominado controlador. Neste dispositivo xéranse ordes moi elementais, microórdes que, sincronizadas polos impulsos do reloxo fan que se vaia executando paso a paso a instrución que está cargada no RI.

# Unidade Aritmético-Lóxica - ALU

Esta unidade é a encargada de realizar as operacións elementais de tipo aritmético (sumas, restas, produtos, divisións) e de tipo lóxico (comparacións).

A UAL recibe os datos da U.C. E tras realizar a operación oportuna devolve o resultado a memoria principal. A maioría das UAL como operación aritmética só teñen a suma e o resto das operacións efectúanas a partir dela.

Consta dos seguintes elementos:



# Unidade Aritmético-Lóxica - ALU

**Circuíto operacional(COP):** Contén os circuítos necesarios para a realización das operacións cos datos procedentes dos rexistros de entrada. Este circuítos ten unhas entradas de ordes para seleccionar a clase de operación que debe realizar en cada momento (suma, resta,...).

**Rexistros de entrada(REN):** Neles almacénanse os datos ou operandos que interveñen nunha instrución antes da realización da operación por parte do circuítos operacional. Tamén se empregan para o almacenamento de resultados intermedios ou finais das operacións respectivas.

**Rexistro acumulador (RA):** Almacena os resultados das operacións levadas a cabo polo circuítos operacional. Está conectado cos rexistros de entrada para realimentación no caso de operacións encadeadas. Ten unha conexión directa ao bus de datos para o envío dos resultados á memoria central ou á unidade de control.

**Rexistro de estado (RES):** Trátase dun conxunto de circuítos nos que se deixa constancia dalgunhas condicións que se deron na última operación realizada e que haberán de terse en conta en operacións posteriores.

# UNIDADE DE COMA FLOTANTE

**Hoxe en día na UCP tamén se inclúe unha FPU**

A **FPU** (Floating Point Unit) é unha unidade de cálculo especial, que funciona con números de coma flotante de varias lonxitudes, segundo a precisión necesaria. É dicir, a **FPU** é unha calculadora que alivia a carga da ALU.

Nos primeiros tempos do PC os microprocesadores non incluían esta unidade, que podía incorporarse en forma de coprocesador matemático, ou ben, realizar os cálculos mediante software ou “emuladores”.

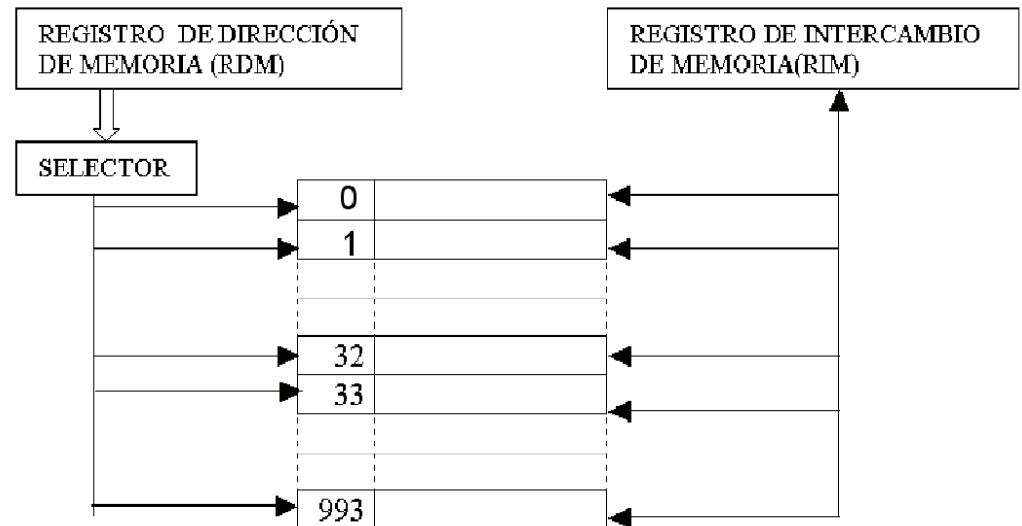
# Memoria principal

A memoria central, principal ou interna é a unidade onde están almacenadas as instrucións e os datos necesarios para poder realizar un determinado proceso. Está constituída por multitude de **celas** ou **posicións de memoria**, numeradas de forma consecutiva, capaces de reter, mentres o ordenador estea acendido, a información depositada nelas.

A numeración das celas denomínase **dirección de memoria** e mediante esta pódese acceder de forma directa a calquera delas independentemente da súa posición; é dicir, é un soporte de información de **acceso directo**.

Ten asociados os seguintes rexistros:

En cada posición de memoria almacenase 1 byte



# Memoria principal

**Rexistro de dirección de memoria (RDM):** Contén a dirección da cela na que se vai a realizar a seguinte operación (lectura ou escritura).

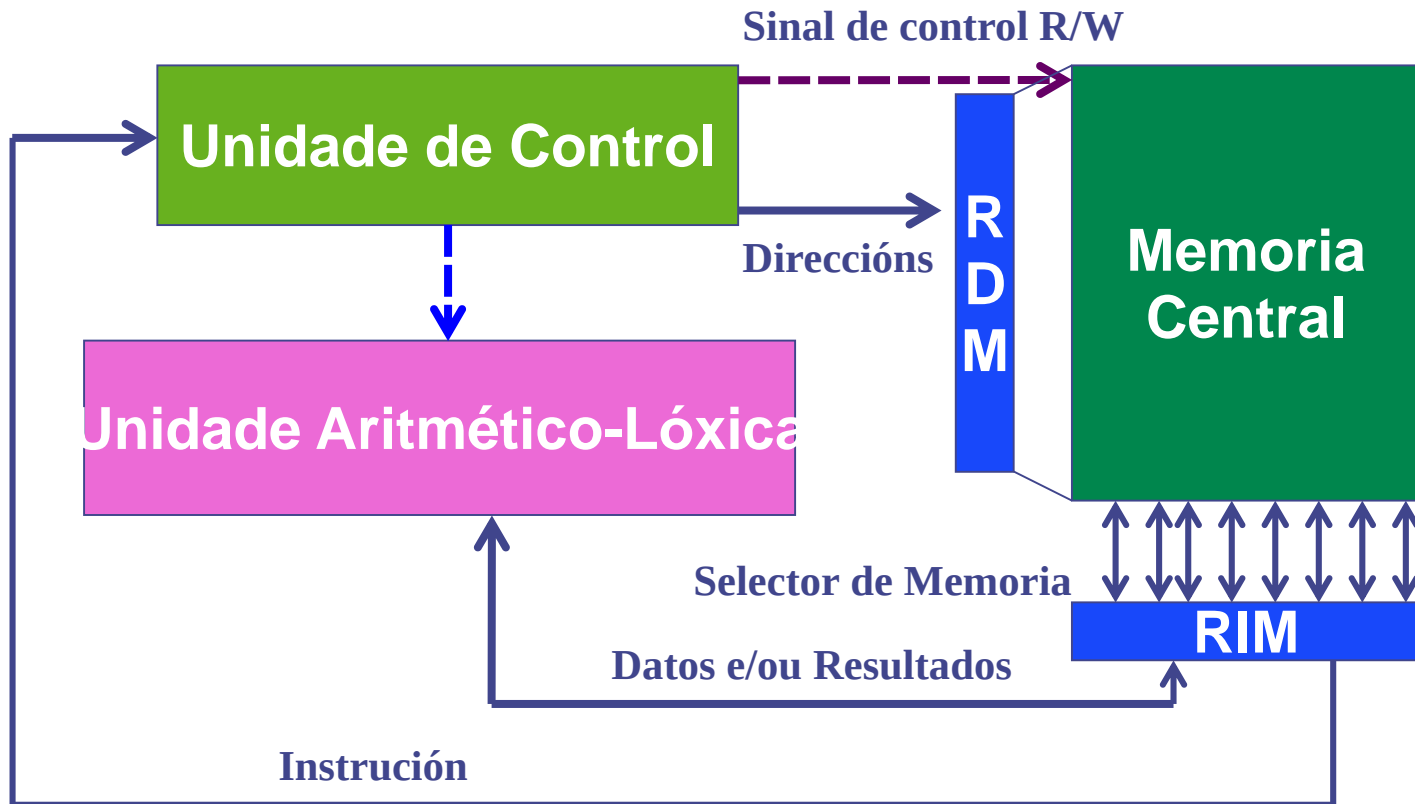
**Rexistro de intercambio de memoria (RIM):**

Se se trata dunha **operación de lectura**, este rexistro é o que recibe o dato da memoria sinalado polo RDM para o seu envío por medio do bus do sistema a unidade que o require.

Se se trata dunha **operación de escritura** contén a información que hai que gravar na posición indicada polo RDM.

**Selector de memoria:** Este dispositivo actívase cada vez que se produce unha orde de lectura ou escritura, conectando a cela de memoria a dirección da cal figura o RDM, co RIM e posibilitando a transferencia dos datos nun sentido ou noutro.

# Procesador e Memoria





# CACHÉ DO PROCESADOR

A memoria caché utilízase para gardar as posicións da memoria principal máis utilizadas. Almacenando a información na caché, increméntase enormemente a velocidade de adquisición dos datos.

Cando a caché contén os datos que necesita a CPU, non hai tempos de espera e denomínase *acerto de caché*. Cando a caché non contén os datos, denomínase *fallo de caché*, e a CPU terá que esperar ata que a memoria principal entregue os datos.

Na práctica existen polo menos dúas memorias caché. Chámanse memoria caché de Nivel 1 (*L1*), de Nivel 2 (*L2*), e se existe de Nivel 3 (*L3*).

# CACHÉ DO PROCESADOR

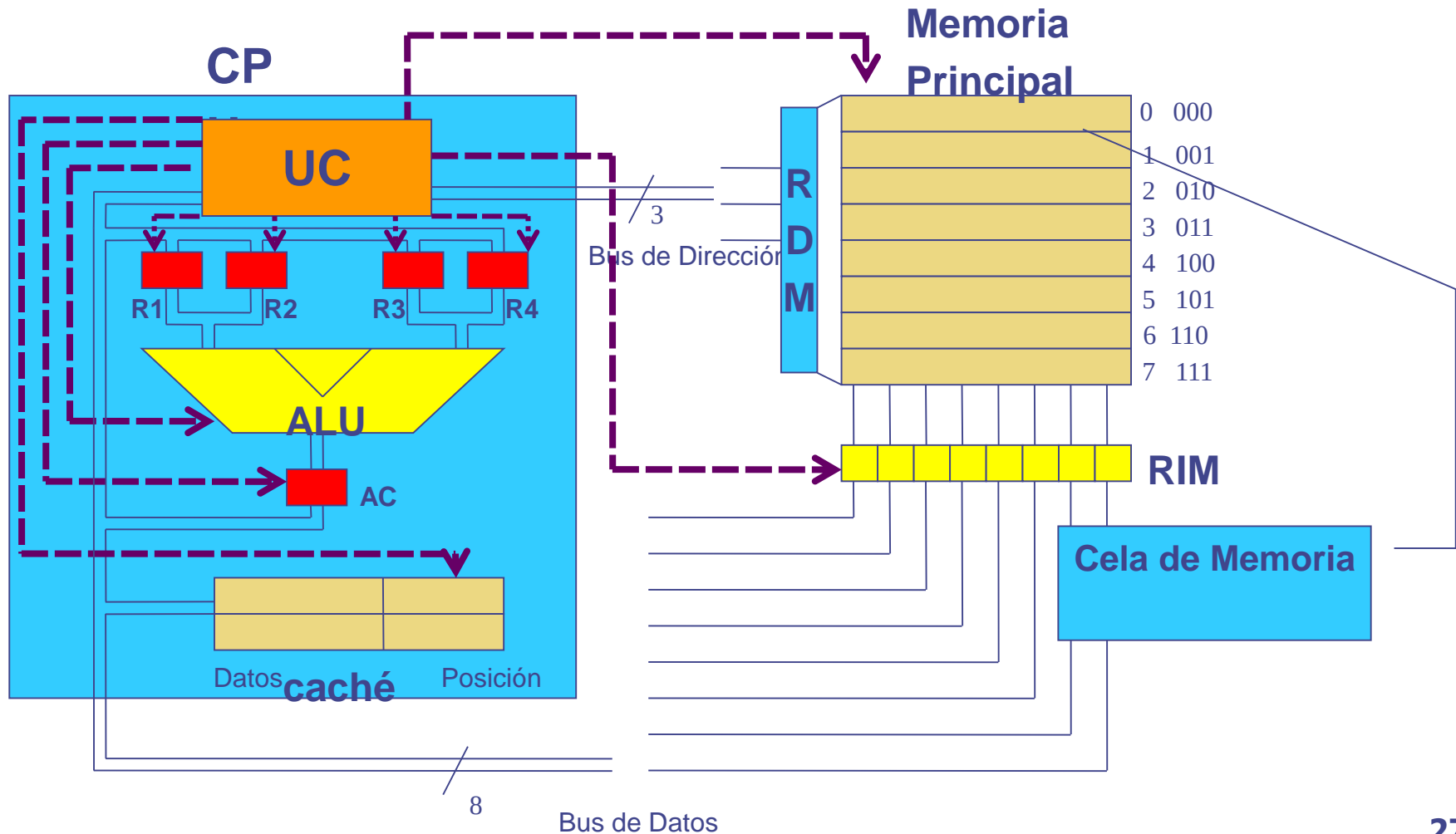
**Pregunta: Se ter a memoria caché preto do procesador é tan beneficioso, ¿por que non se utiliza memoria caché para toda a memoria principal?**

Por unha sinxela razón, a memoria caché xeralmente utiliza un tipo de chip de memoria chamado SRAM (RAM estático), que é máis caro e require máis espazo físico por megabyte que o DRAM, que normalmente se utiliza para a memoria principal.

Así mesmo, mentres a memoria caché mellora o desempeño xeral do sistema, isto faise ata certo punto, o beneficio real da memoria caché é almacenar as instrucións que se utilizan con máis frecuencia. Unha memoria caché máis grande mantería máis datos, pero se iso non se necesita con frecuencia, non hai un grande beneficio en telo xunto ao procesador.

# CPU E MP

Ver exemplo ejecución instrucción



# BUS

O **bus** é o elemento de comunicación entre os diferentes compoñentes do ordenador. Fisicamente a súa descrición é un conxunto de fíos físicos utilizados para a transmisión de datos entre os compoñentes dun sistema informático.

Os buses caracterízanse por:

- **Ancho** O número e disposición das súas liñas (cada liña transmite un bit)
- **Frecuencia** – Médese en Hz – Número de paquetes de datos que poden ser enviados ou recibidos por segundo
- **Velocidade de transferencia** – Cantidade de datos que se poden transmitir por unidade de tempo.

# BUS

**O bus pode ser un cuello de botella.** Os buses están constituídos por entre 50 e 100 liñas distintas que a súa vez divídense nos seguintes **subconxuntos**.

- **Bus de datos.** Transmite información entre a CPU e resto das unidades
- **Bus de direccións.** Contén a dirección do destino (da memoria ou E/S) ao que van dirixidos os datos que se están transmitindo polas liñas de datos.
- **Bus de control.** Mediante as liñas de control transmítense as ordes procedentes da unidade de control ás outras unidades.

Algúns deseños utilizan **liñas eléctricas multiplexadas** para o bus de dirección e o bus de datos. Isto significa que un mesmo conxunto de liñas eléctricas compórtanse unhas veces como bus de direccións e outras veces como bus de datos, pero nunca ao mesmo tempo. Unha liña de control permite discernir cal das dúas funcións está activa.

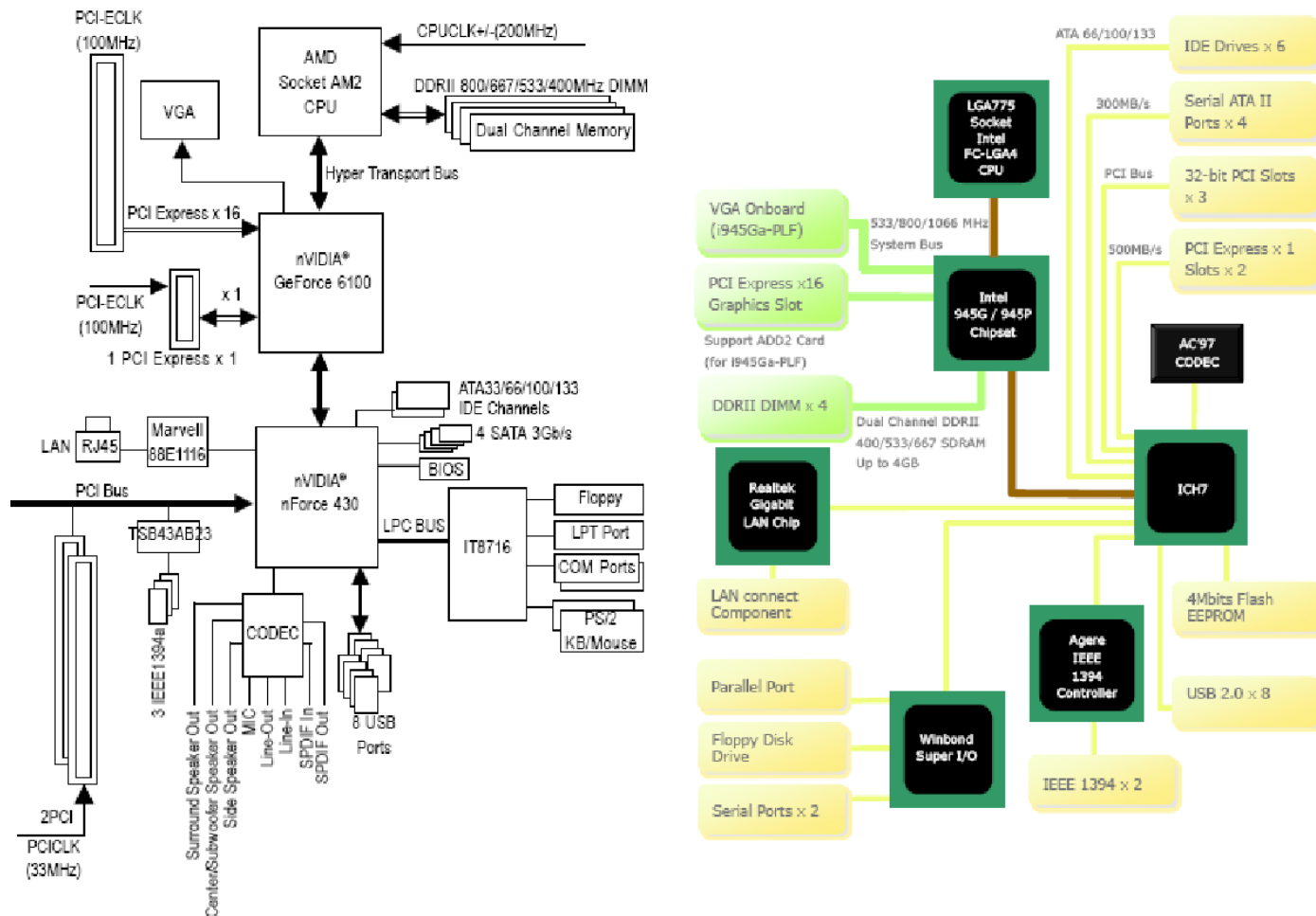
# BUS

**Bus do sistema** - Hai un único bus compartido que interconecta tanto ao procesador e á memoria principal coma a todos os dispositivos de entrada/saída. A este bus coñéceselle co nome de **bus do sistema**. Cando se fala do ancho do bus de datos e o ancho do bus de direccións referímonos o ancho deste bus. Antigamente tiñamos o Bus Frontal do sistema (FSB) actualmente este bus foi substituído por Intel QuickPath Interconnect e posteriormente DMI en Intel e Hypertransport e posteriormente UMI en AMD

**Buses de E/S.** Os dispositivos de entrada/saída comunícanse entre si a través dun bus dedicado que recibe o nome de bus de entrada/saída. Como os dispositivos de entrada/saída requiren estar conectado coa memoria principal, habilítase un acceso a esta dende o bus de entrada/saída mediante un dispositivo chamado adaptador de bus. Esta organización libera tráfico entre o procesador e a memoria, separando as transaccións de entrada/saída.

Independentemente da velocidade do bus do sistema temos a velocidade interna do microprocesador. Esta velocidade é moito maior ca as anteriores e é a que se adoita publicitar nun ordenador

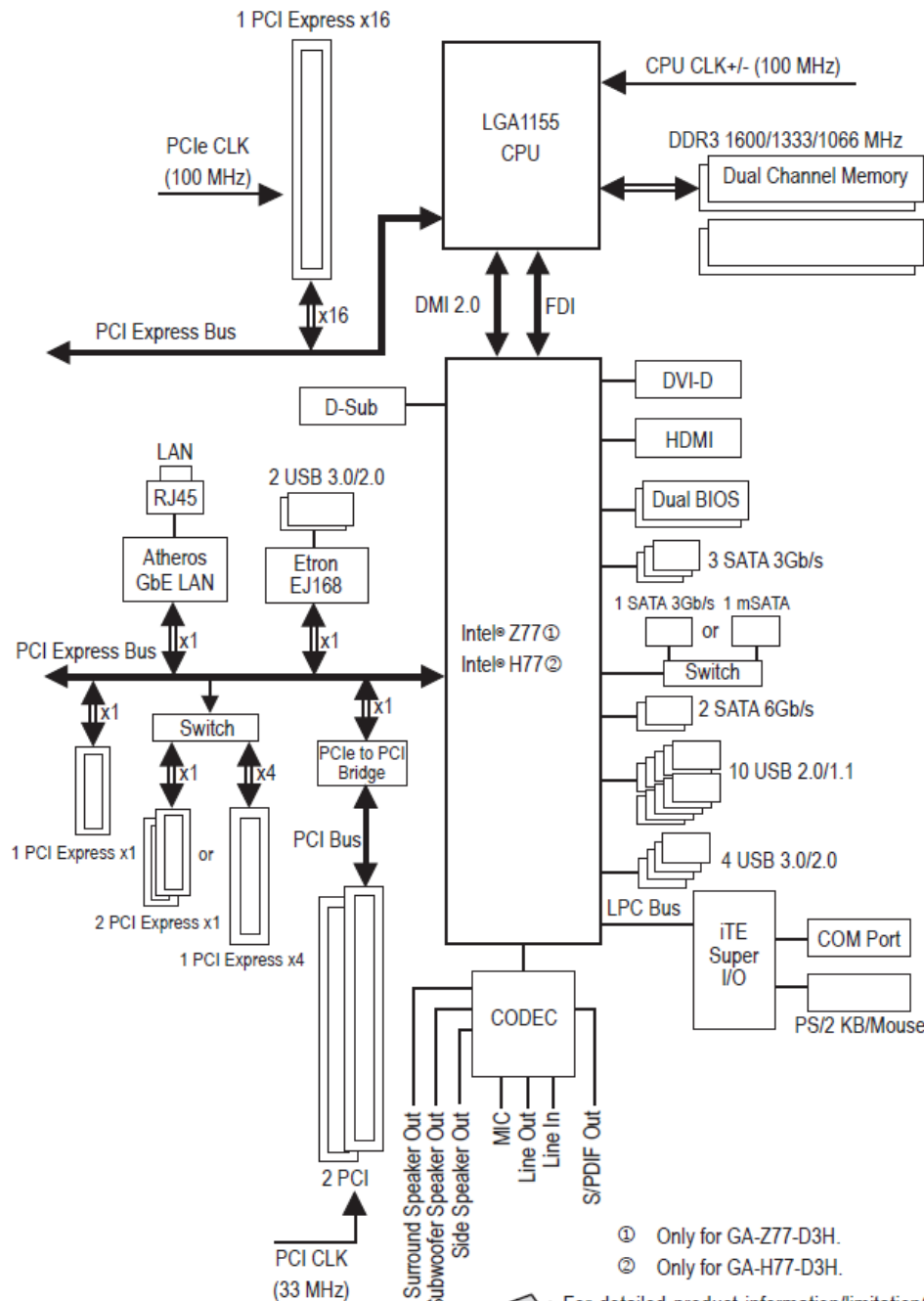
# Placas base antigas



En algunhas arquitecturas, a memoria RAM comunícase directamente co procesador sen pasar polo Northbridge

- En AMD esta tecnoloxía chámase **HyperTransport**
- En Intel **Intel QuickPath Interconnect** (dispoñible a partir de socket 1156) e **DMI**

## GA-Z77-D3H/GA-H77-D3H Motherboard Block Diagram



## Placa moderna Intel

Desaparece FSB

Desaparece Intel Quick Path Interconnect

Desaparece Chipset Norte e Chipset Sur

Nesta placa de Intel desaparece o chipset sur. Quedando un único Chip chamado **(PCH)** - **Platform Controller Hub**

Ten a tecnoloxía:

**DMI** (Direct Media Interface) - Mellora de Intel Quick Path Interconnect

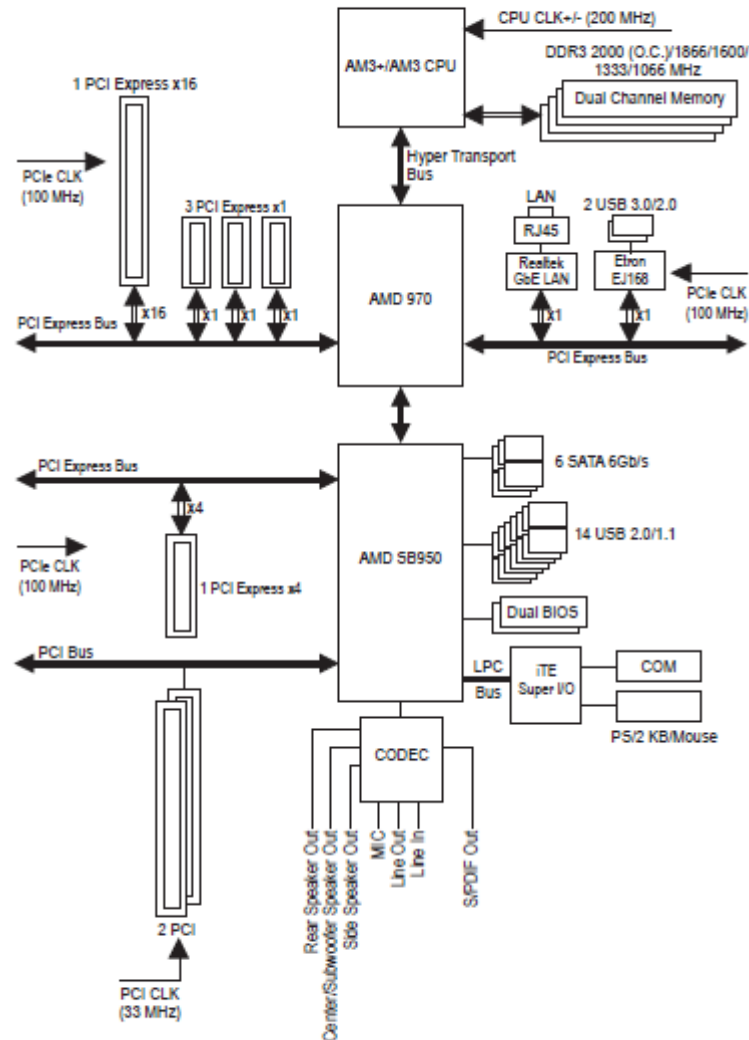
**FDI** - FDI(Flexible Display Interface). Este enlace surge do feito de ter unha tarxeta gráfica Integrada no propio procesador. Agora necesitamos levar os gráficos dende este elemento a pantalla. Esta liña leva os datos dende o procesador as partes do chipset encargadas de darlle servizo.

DMI 1.0 transfere 10 Gbps  
DMI 2.0 foi introducida en 2011 e transfere 20 Gbps



# Placa Base Moderna AMD

GA-970A-D3 Motherboard Block Diagram



For detailed product information/limitation(s), refer to "1-2 Product Specifications."

# Medidas de Información

No **sistema binario** so existen dous símbolos diferentes: 1 e 0. Unha información que só pode tomar como valores o 0 e o 1 denomínase **bit** e forma a unidade básica de información.

Un ordenador, debido á súa construción baseada fundamentalmente en circuítos electrónicos dixitais, traballa co sistema binario (1 , 0). Este é o motivo que nos obriga a transformar internamente todos os nosos datos, tanto numéricos como alfanuméricos, a unha representación binaria para que a máquina sexa capaz de procesalos.

# Medidas de Información Capacidade

**Byte:** Agrupación de 8 bits.

1 KB(kilobyte) = 1000 bytes =  $10^3$  bytes

1 MB(megabyte) = 1000 KB =  $10^6$  bytes

1 GB(gigabyte) = 1000 MB =  $10^9$  bytes

1 TB(terabyte) = 1000 GB =  $10^{12}$  bytes

1 PB(petabyte) = 1000 TB =  $10^{15}$  bytes

1 EB(exabyte) = 1000 PB =  $10^{18}$  bytes

1 ZB(zettabyte) = 1000 EB =  $10^{21}$  bytes

1 YB(yottabyte) = 1000 ZB =  $10^{24}$  bytes

**Byte:** Agrupación de 8 bits.

1 KiB(kibibyte) = 1024 bytes =  $2^{10}$  bytes

1 MiB(mebibyte) = 1024 KiB =  $2^{20}$  bytes

1 GiB(gibibyte) = 1024 MiB =  $2^{30}$  bytes

1 TiB(tebibyte) = 1024 GiB =  $2^{40}$  bytes

1 PiB(pebibyte) = 1024 TiB =  $2^{50}$  bytes

1 EiB(exbibyte) = 1024 PiB =  $2^{60}$  bytes

1 ZiB(zebibyte) = 1024 EiB =  $2^{70}$  bytes

1 YiB(yobibyte) = 1024 ZiB =  $2^{80}$  bytes

# Medidas de Información

## Capacidade

¡Problema! Ata fai uns anos (1999), non existía os KiB, MiB, GiB... Tampouco se utilizaban as potencias de dez cando se medía a información.

Antes:

**1 KB = 1024 bytes.**

**1 MB = 1024 KB bytes.**

...

Isto hoxe en día non debería ser correcto. A transparencia anterior é como se ten que nomear e facer as conversións segundo os estándares. Pero... moitos fabricantes e comerciantes utilizan aínda a nomenclatura incorrecta

# Medidas de Información

## Capacidade

**Solución: Coñecer o problema.**

1. Cando un fabricante che fala de 4 GB de memoria RAM, se refire a 4 GiB.
2. Cando un fabricante di que a caché é de 256 KB, quere dicir que ten 256 KiB.
3. Cando un fabricante di que un disco duro ten 500 GB, quere dicir que ten  $500 \text{ GB} = 500 * 1000 \text{ MB}$ .
4. En xeral, capacidades de memorias principais sempre se utilizan múltiplos de 1024 e capacidades de memorias secundarias se utilizan múltiplos de 1000.

# Medidas de Información

## Capacidade

### Medida formal para a capacidade

Unidades de información (del Byte)			
Sistema Internacional (Decimal)		ISO/IEC 80000-13 (Binario)	
Múltiplo - (Símbolo)	SI	Múltiplo - (Símbolo)	ISO/IEC
kilobyte (kB)	$10^3$	Kibibyte (KiB)	$2^{10}$
Megabyte (MB)	$10^6$	Mebibyte (MiB)	$2^{20}$
Gigabyte (GB)	$10^9$	Gibibyte (GiB)	$2^{30}$
Terabyte (TB)	$10^{12}$	Tebibyte (TiB)	$2^{40}$
Petabyte (PB)	$10^{15}$	Pebibyte (PiB)	$2^{50}$
Exabyte (EB)	$10^{18}$	Exbibyte (EiB)	$2^{60}$
Zettabyte (ZB)	$10^{21}$	Zebibyte (ZiB)	$2^{70}$
Yottabyte (YB)	$10^{24}$	Yobibyte (YiB)	$2^{80}$
Véase también: Nibble · Byte · Octal			

Pasar 24756 bits a KiB e KB

$24756 / 8 / 1000 = 3,072$  KB

$24756 / 8 / 1024 = 3$  KiB

Pasar 6291456 bytes a MiB e MB

$6291456 / 1024 / 1024 = 6$  MiB

$6291456 / 1000 / 1000 = 6,29$  MB

# Exemplo

Supoñemos que en Windows instalamos un Ubuntu nunha máquina en VirtualBox con un disco de 20 GB.

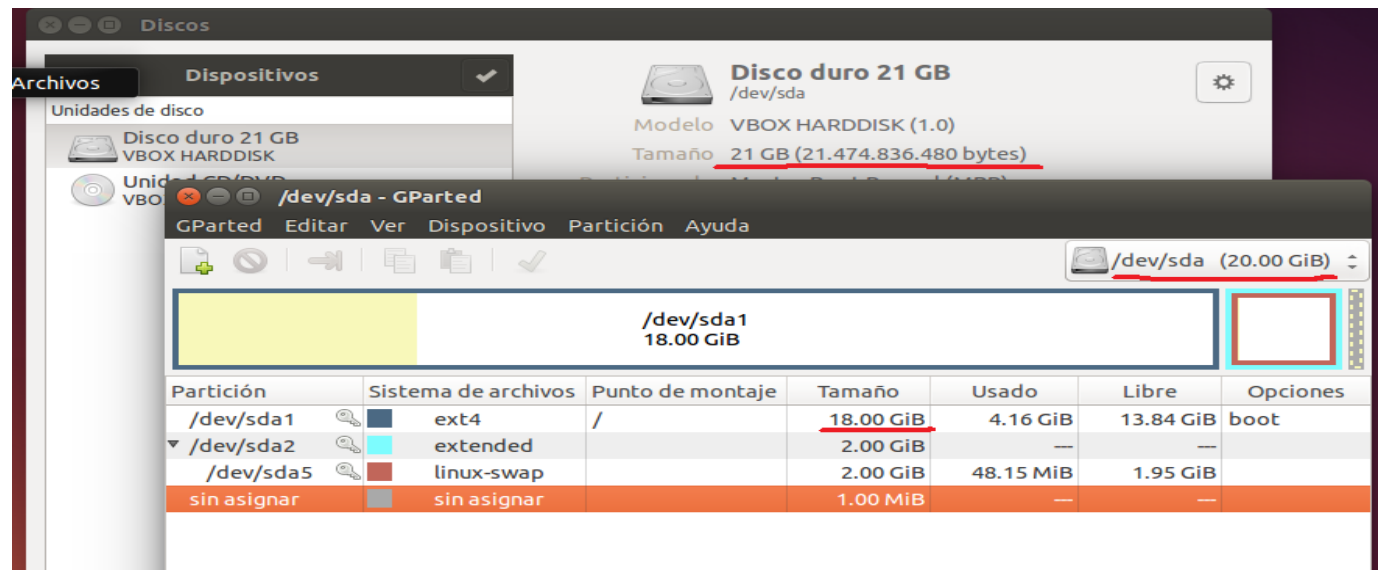
Baixo Windows 20 GB =  $20 \cdot 1024 \cdot 1024 \cdot 1024 = 21474836480$  bytes.

Se se fixera nunha máquina real 20 GB serían  $20 \cdot 1000 \cdot 1000 \cdot 1000 = X$  bytes

Windows só utiliza un sistema binario para os GB non utiliza a medida GiB

Linux utiliza un sistema binario para os GiB e un sistema decimal para os GB.

Disco  
recoñecido  
en Linux



# Medidas de Información

## Capacidade

Engaño na capacidade dos discos

**Cal é a capacidade real de un disco duro SATA publicitado de 500 GB**

**Os fabricantes fan a trampa de pasar dunha medida a outra realizando multiplicacións/divisións por/entre 1000 en vez de 1024**  
 **$500\text{GB} \times 1000 \times 1000 \times 1000 = 5000000000000$  bytes**

**Pero en realidade o S.O. Windows Recoñece como**  
 **$5000000000000 / 1024 / 1024 / 1024 = 465,66$  GB ou máis formalmente**  
**465,66 GiB**

**Debese ter en conta:**

- Segundo o fabricante a cantidade do disco pode variar lixeiramente e para aforrar costes sóese redondear a baixa.
- A capacidade expresada con prefixo decimal resulta nunha cifra maior que si se expresara con prefixo binario.
- Canto maior capacidade ten un disco duro, maior é a discrepancia entre as cifras que expresan esta capacidade con prefixo decimal e binario



# Medidas de Información

## Frecuencia dun Bus e transferencia

A frecuencia dun bus mídese Hz (ciclos/s).

1 Hz = 1 ciclo/seg

1 KHz = 1000 Hz

1 MHz = 1000 KHz

Existen **buses que en cada ciclo fan varias transferencias** (unha, dúas, tres...). Hoxe en día os fabricantes poden utilizar outra unidade para indicar a velocidade dun bus. Son as T/s ou Transferencias/seg.

1 MT/s =  $10^6$  transferencias/seg

1 GT/s =  $10^9$  transferencias/seg

Ex: Se un bus traballa a 100 MHz e fai 4 transferencias en cada ciclo, entón:  
 $100 \text{ Mciclos/s} \times 4 \text{ trasferencias/ciclo} = 400 \text{ MT/s}$

Para calcular o ancho de banda téñenme que dicir o ancho do bus sempre.

# Medidas de Información

## Taxa de transferencia ou Ancho de banda

É a velocidade á que se transmiten os datos por unha canle, tamén se pode denominar bit rate

**Tasa de Transferencia = Ancho bus \* Velocidade do bus (frecuencia)**

Mídese en información partido segundo, polo tanto:

1 bps = 1 bit/seg = 1 b/seg

1 Byte/seg = 1 B/seg = 8 bits/seg

1 KB/seg = 1000 Byte/seg

1 MB/seg = 1000 KB/seg = 1000000 B/seg

**Problema:** Se teño un bus que ten un ancho de 32 bits, e traballa a 200 Hz e fai **unha transferencia/ciclo**, quere dicir que fai 200 transferencias de datos por segundo, para calcular o ancho de banda:

$200 \text{ ciclos/seg} * 32 \text{ bits/transferencia} * 1 \text{ transferencia/ciclo} * 1 \text{ byte/8 bits} = 800 \text{ B/seg}$

# Medidas de Información

## Direccionamiento

**Número de direcciones de memoria capaz de gestionar.**

**Depende to tamaño do bus de direcciones. Así con un tamaño de X bits podemos direccionar  $2^x$ .**

Por exemplo con 32 bits podemos direccionar  $2^{32} = 4 \text{ GB}$

# Medidas de Información

## Velocidade de procesamiento

A velocidade de procesamiento mídese en Hz e múltiplos do Hz. Indica o número de ciclos partido segundo ao que traballa o micro. Cada instrución do xogo de instrucións que entende o procesador lle leva unha serie de ciclos executala. Haberá instrucións que con dous ciclos xa se executan, outras tres, outras catro...

$$1 \text{ Hz} = 1 \text{ ciclo/seg}$$

$$1 \text{ KHz} = 1000 \text{ Hz}$$

$$1 \text{ MHz} = 1000 \text{ KHz}$$

# Medidas de Información

## Velocidade de procesamiento

**Problema:** Se un microprocesador ten un frecuencia de 300 KHz e un programa ten 100 instrucións das cales o 25% se executan nun só ciclo, un 50% necesitan dous ciclos para executarse e o resto necesitan 5 ciclos para executarse, ¿cantos ciclos de microprocesador necesita o programa para executarse enteiro? ¿canto tempo lle leva?

$25 \text{ instrucións} * 1 \text{ ciclo/instrución} + 50 \text{ instrucións} * 2 \text{ ciclos/instrución} + 25 \text{ instrucións} * 5 \text{ ciclos/instrución} = 250 \text{ ciclos}$

$300 \text{ KHz} = 300 * 1000 \text{ ciclos/seg} \Rightarrow 1/300000 \text{ seg/ciclo}$

$250 \text{ ciclos} * 1/300000 \text{ seg/ciclo} = 0,00083333 \text{ seg}$

# Medidas de Información

## Rendemento dun computador

**FLOPS:** Float Point Operation per Second – Operacións en punto flotante por segundo.

Medida do rendemento dun computador, especialmente en cálculos científicos que usan gran número de operacións en coma flotante.

$$1 \text{ MFLOPS} = 10^6 \text{ FLOPS}$$

$$1 \text{ GFLOPS} = 10^9 \text{ FLOPS}$$

$$1 \text{ TFLOPS} = 10^{12} \text{ FLOPS}$$

...

OLLO!!! Non se debe tomar como única medida para valorar a capacidade dun computador.

# Sistemas de Codificación

# Sistemas de codificación numérica

Sistema de numeración é o conxunto de símbolos e regras que se utilizan para representar cantidades e datos numéricos. Estes sistemas de codificación numérica caracterízanse pola base

- **Binario** - Utiliza dous símbolos diferentes cero e un – (É o sistema que utiliza o ordenador)
- **Octal** - Sistema en base 8 que utiliza os números do 0 ao 7
- **Decimal** - Sistema en base 10. Utiliza os números do 0 ao 9
- **Hexadecimal** - Sistema de numeración en base 16 utiliza os números do 0 ao 9 e as letras da A á F



# Conversión de calquera base a decimal (TFN)

O Teorema fundamental da Numeración (TFN) Serve para relacionar unha cantidade expresada en calquera sistema de numeración coa mesma cantidade expresada no sistema decimal

$i$  – posición do dígito con respecto o punto decimal

$X_i$  – Valor absoluto do dígito

$B$  – Base

Pódese expresar como:  $\sum X_i * B^i$

Un número en base 10 pódese expresar como:

$$\dots X_2 * 10^2 + X_1 * 10^1 + X_0 * 10^0 + X_{-1} * 10^{-1} + X_{-2} * 10^{-2} \dots$$

**Exemplo:** pasar  $1101001.01_{(2)}$  a decimal

$$1 * 2^6 + 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 + 0 * 2^{-1} + 1 * 2^{-2} = 105,25$$

# Conversión de cualquier base a decimal (TFN)

Sexa o número en octal 3714 convertelo a decimal

$$3 * 8^3 + 7 * 8^2 + 1 * 8^1 + 4 * 8^0 = 1996$$

entón  $3714_{(8)} = 1996_{(10)}$

O número en hexadecimal 1B7F convertelo a decimal

1	B	7	F
1	11	7	15

$$1 * 16^3 + 11 * 16^2 + 7 * 16^1 + 15 * 16^0 = 1 * 4096 + 11 * 256 + 7 * 16 + 15 * 1 = 7039$$

e dicir,  $1B7F_{(16)} = 7039_{(10)}$

# Conversión de Decimal a outra base

**Pasar o número  $285,75_{(10)}$  a binario**

**Divídese o número en base 10 sucesivamente entre a base á que se quere pasar o número**

$$\begin{array}{r} 285 \quad | \quad 2 \\ 08 \quad 142 \\ 05 \quad 02 \quad | \quad 2 \\ 1 \quad 0 \quad 71 \\ 11 \\ 1 \quad 35 \\ 15 \quad | \quad 2 \\ 17 \\ 8 \\ 0 \end{array}$$

**Cóllese o último cociente e os restos empezando polo final**

$$285_{(10)} = 100011101_{(2)}$$

$$\begin{array}{r} | \quad 2 \\ 4 \\ 0 \end{array} \quad \begin{array}{r} | \quad 2 \\ 2 \\ 0 \end{array} \quad \begin{array}{r} | \quad 2 \\ 1 \end{array}$$

## **Se o número ten parte decimal**

Multiplicamos repetidamente a parte decimal pola base ata que esta sexa 0. A parte enteira de cada multiplicación formará os bits de número binario.

Collémoslos dende o principio

$$0,75 * 2 = 1,5$$

$$0,5 * 2 = 1,0$$

$$285,75_{(10)} = 100011101,11_{(2)}$$

# Conversión de Decimal a outra base

Sexa o número decimal 105,25 convertelo a binario

105	2						
05	52	2					
1	12	26	2				
	0	06	13	2			
		0	1	6	2		
				0	3	2	
					1	1	2
						1	

$$105_{10} = 1101001_2$$

A parte decimal é 0,25

$$0,25 * 2 = 0,50$$

$$0,50 * 2 = 1,0$$

$$0,25_{10} = 0,01_2$$

$$105,25_{10} = 1101001,01_2$$

# Conversión de decimal a outra Base

Sexa o número decimal 782 convertelo a octal

$$\begin{array}{r|l} 782 & 8 \\ \hline 062 & 97 \\ 06 & 17 \\ & 1 \end{array} \quad \begin{array}{r|l} & 8 \\ \hline & 12 \\ & 04 \end{array} \quad \begin{array}{r|l} & 8 \\ \hline & 1 \end{array}$$

$$782_{(10)} = 1416_8$$

Pasar  $5492_{(10)}$  a hexadecimal

$$\begin{array}{r|l} 5492 & 16 \\ \hline 069 & 343 \\ 052 & 023 \\ 04 & 07 \end{array} \quad \begin{array}{r|l} & 16 \\ \hline & 21 \\ & 05 \end{array} \quad \begin{array}{r|l} & 16 \\ \hline & 1 \end{array}$$

$$\text{Entón, } 5492_{(10)} = 1574_{(16)}$$

# Sistema Decimal, Binario, Octal e Hexadecimal

DECIMAL	BINARIO	OCTAL	HEXADECIMAL
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

# Transformación directa (Dunha base $2^n$ a binario)

Se queremos realizar unha transformación directa entre un número en base  $2^n$  a binario. Collemos individualmente cada dígito do número e calculamos a súa correspondencia en binario, utilizando para iso **n** bits

Dado o número  $65470_{10}$  convertelo a binario

$$\begin{array}{ccccc} 6 & 5 & 4 & 7 & 0 \\ \hline 110 & 101 & 100 & 111 & 000 \end{array}$$

$8 = 2^3$  Por iso representase o binario con 3 bits

Conservando a orde do número octal, o binario equivalente é: es:  
 $65470_{10} = 110\ 101\ 100\ 111\ 000_{2}$

Converter o número  $7AE90_{16}$  a binario

$$\begin{array}{ccccc} 7 & A & E & 9 & 0 \\ \hline 0111 & 1010 & 1110 & 1001 & 0000 \end{array}$$

$16 = 2^4$  Por iso representase o binario con 4 bits

$7AE90_{16} = 111\ 1010\ 1110\ 1001\ 0000_{2}$

# Transformación directa (De binario a unha base $2^n$ )

Se queremos realizar unha transformación directa entre un número en binario a base  $2^n$ . Collemos grupos de  $n$  bits do número binario e calculamos o seu correspondente valor na base  $2^n$

Exemplo. Converter o número  $11100011101001_{12}$  a octal.

$$\begin{array}{ccccccccc} 11 & 100 & 011 & 101 & 001 \\ \hline 3 & 4 & 3 & 5 & 1 \end{array}$$

$$11100011101001_{12} = 34351_8$$

$8 = 2^3$  Por iso cóllense grupos de 3 bits

Converter  $111000101001101_{12}$  a hexadecimal.

$$\begin{array}{ccccccc} 111 & 0001 & 0100 & 1101 \\ \hline 7 & 1 & 4 & D \end{array}$$

$$111\ 0001\ 0100\ 1101_{12} = 714D_{16}$$

$16 = 2^4$  Por iso cóllense grupos de 4 bits



# Transformación Indirecta

Para o resto dos cambios de base debemos utilizar o método indirecto que consiste en pasar o número de base  $n$  a base 10 e posteriormente a base  $m$

**Pasar o número  $34_5$  a base 8**

**Primeiro pasase a decimal  $\rightarrow 3 * 5^1 + 4 * 5^0 = 19_{10}$**

**Despois pasase a octal**

$$\begin{array}{r|l} 19 & 8 \\ 03 & 2 \end{array}$$

**entón,  $34_5 = 19_{10} = 23_8$**

# Resume de cambios de base

Base Orixe	Base Destino	Método
Calquera Base	Decimal	<p><math>X_i</math> –Díxito valor absoluto. B-Base, i posición do díxito (empeza en 0 de dereita a esquerda para a parte enteira e en –1 de esquerda a dereita para a parte decimal).</p> $\sum X_i * B^i$
Decimal	Calquera base	Dividir sucesivamente entre a base, quedar co último cociente e os restos en orde inversa. Para a parte decimal multiplicamos sucesivamente pola base a parte decimal ata que esta sexa cero. Quedamos coas partes enteiras
$2^n$	Binario	Representar en binario cada díxito do número utilizando <b>n</b> bits.
Binario	$2^n$	Coller grupos de n bits (de dereita a esquerda na parte enteira e de esquerda a dereita na parte decimal) e representar o número na base $2^n$ . Completar con 0 pola esquerda a parte enteira e pola dereita a decimal
Calquera Base	Calquera base	Pasar o número da base orixe a base 10 e de base 10 á base destino

# Aritmética binaria.

+	0	1
0	0	1
1	1	0 <sup>(1)</sup>

-	0	1
0	0	1 <sup>(1)</sup>
1	1	0

x	0	1
0	0	0
1	0	1

# Lógica binaria.

or	0	1
0	0	1
1	1	0 <sup>(1)</sup>

NOT	
0	1
1	0

and &	0	1
0	0	0
1	0	1

\*Tenemos más: XOR, XNOR...

# Representación numérica.

## BCD (Código Decimal Binario).

- Representase cada cifra do número por n bits:
- 35 con 4 bits: 0011 0101.

## Enteros.

- Signo magnitud (SM).
  - S: bit para signo (0 positivo, 1 negativo).
  - M: Resto de bits para valor (magnitud).
    - -35 con 8 bits:

1	0100011
S	M

# Representación numérica.

- Complemento a 1.
  - Bit para signo (0 positivo, 1 negativo).
  - Positivos en binario puro.
  - Negativos, cambiamos 0 por 1 y viceversa.
    - -35 con 8 bits: 11011100
- Complemento a 2.
  - Positivos en binario puro.
  - Negativos, calculamos complemento a 1 y sumamos 1 al resultado. La suma se realiza sin acarreos.
    - -35 con 8 bits: 11011101.

# Representación numérica.

Exceso a  $2^n-1$ .

- Sin bit de signo.
- Representación:
  - Número + exceso. Calculamos valor de resta en binario puro.
  - Número: el que vamos a representar.
  - Exceso:  $2^{n-1}$
  - -35 con 8 bits:  $n=8 \rightarrow 2^{n-1} = 2^{8-1} = 128 \rightarrow -35 + 128 = 93 = 01011101$ .

# Representación numérica.

## Punto flotante.

- Tres campos: signo (1 bit), mantisa y exponente:  $(-1)^{\text{signo}} \times \text{mantisa} \times 2^{\text{exponente}}$ 
  - **Signo:** 0 positivo y 1 negativo.
  - **Mantisa:** Normalizar para obtener un número en coma fija de forma 0.XXXXXX y se almacena sólo la parte fraccionaria (XXXXXX).
  - **Exponente:** Puede ser un valor positivo o negativo.  
Representación en complemento a 2:
    - Con n bits, codificamos en binario puro el resultado de:  
 $\text{bits\_exponente} = \text{valor\_exponente} + (2^{n-1} - 1)$ .
    - A partir del código del exponente podemos obtener su valor:  
 $\text{valor\_exponente} = \text{bits\_exponente} - (2^{n-1} - 1)$ .

# Representación numérica.

## Punto flotante.

- Ejemplo: -35 con 1 bit de signo, 5 bits exponente y 8 bits mantisa:
  - 35 en binario puro: 0100011
  - Normalizar:  $0100011 = 0100011 * 2^0 = 1,00011 * 2^5$
  - **Mantisa** (con bit implícito): 00011
  - **Exponente**:  $5 + 2^{5-1} - 1 \rightarrow 5 + 15 \rightarrow 20 \rightarrow 10100$
  - **Signo**: 1

Signo	Mantisa	Exponente
1	00011000	10100

- Estándar IEEE 754. Define formatos (número de bits).

Precisión	bits	signo	mantisa	exponente
Simple precisión	32	1	23	8
Doble precisión	64	1	52	11
Precisión extendida	80	1	64	15



# Codificación alfanumérica

Para representar datos alfanuméricos necesitamos outros sistemas de codificación chamados Sistemas de Codificación Alfanuméricos.

Os sistemas de codificación alfanumérica serven para representar unha cantidade determinada de símbolos, en binario. A cada símbolo corresponderalle unha combinación dun número de bits.

A asignación de códigos é arbitraria, e polo tanto cada fabricante podería asignar unha combinación diferente ao mesmo carácter. Para combater o caos que iso provocaría, créanse códigos que normalicen esta situación, e que se aceptan entre toda a comunidade informática como estándares.

Os sistemas de codificación alfanumérica mais utilizados son:  
**EBCDIC, ASCII e UNICODE**

# Codificación alfanumérica

■ **EBCDIC** – Foi o primeiro que se ideou como código interno de ordenadores. Creado por IBM para utilizalo nos seus ordenadores. Aínda que non é moi utilizado en microcomputadoras, sí é coñecido e aceptado internacionalmente, sendo utilizado principalmente como código de IBM para mainframes (grandes computadoras) e minicomputadoras.

Utiliza 8 bits co que é capaz de representar 256 caracteres.

■ **UNICODE** – Código estándar internacional que se utiliza na maioría dos S.O. actuais e navegadores de Internet. Permite que un produto software estea dispoñible para varias plataformas, idiomas ou países, sen necesidade de modificar o seu deseño.

Existe unha táboa única para todos os países. Temos UTF-8 (de 1 a 4 bytes), aínda que o máis utilizado é UTF-16 (de 16 a 24 bits), tamén hai UTF-32 (Os primeiros 128 caracteres son iguais os ascii)

# Codificación alfanumérica

■ **ASCII** – Código estándar estadounidense.

Pódense representar  $2^7$  é dicir 128 símbolos no ASCII standard e  $2^8$  (256) símbolos distintos no ASCII extendido.

O ASCII extandard é universal sen embargo o extendido incorpora caracteres específicos para cada país xa que os diferentes símbolos non caberían nunha única táboa.

Os caracteres a partir do 128 poden ser distintos entre países e sistemas operativos. Na antigüidade os sistemas operativos utilizaban esta codificación.

Hoxe en día séguense utilizando as variantes ISO-8859-1 (windows-1252) - latín 1 – norma ISO que define a codificación do alfabeto latino incluíndo letras acentuadas e especiais (como ß, Ø), necesarios para a escritura das linguas de Europa occidental: afrikáans, alemán, aragonés, asturiano, catalán, danés, escocés, español, feroés, finés, francés, gaélico, gallego, inglés, islandés, italiano, neerlandés, noruego, portugués, sueco e Euskera.

# EBCDIC

Cada símbolo represéntase mediante unha combinación de 8 bits, agrupados en dous bloques: 4 bits de zona e 4 bits de díxito. Por exemplo, o carácter “a” representaría-se:

Bits de zona	Bits de díxito
1000	0001

Código EBCDIC

izda. → dcha. ↓	0000 0	0001 1	0010 2	0011 3	0100 4	0101 5	0110 6	0111 7	1000 8	1001 9	1010 A	1011 B	1100 C	1101 D	1110 E	1111 F
0000 0	NUL	DLE	DS		SP	&	-						{	}	\	0
0001 1	SOH	DC1	SOS			/			a	j			A	J		1
0010 2	STX	DC2	FS	SYN					b	k	s		B	K	S	2
0011 3	ETX	TM							c	l	t		C	L	T	3
0100 4	PF	RES	BYP	PN					d	m	u		D	M	U	4
0101 5	HT	NL	LF	RS					e	n	v		E	N	V	5
0110 6	LC	BS	ETB	UC					f	o	w		F	O	W	6
0111 7	DEL	IL	ESC	EOT					g	p	x		G	P	X	7
1000 8		CAN							h	q	y		H	Q	Y	8
1001 9	RLF	EM							i	r	z		I	R	Z	9
1010 A	SMM	CC	SM		cent	!	!	:								
1011 B	VT	CU1	CU2	CU3	.	\$	.	#								
1100 C	FF	IFS		DC4	<	*	%	@								
1101 D	CR	IGS	ENQ	NAK	(	)	—	'								
1110 E	SO	IRS	ACK		+	:	>	=								
1111 F	SI	IUS	BEL	SUB		→	?	”								

# ASCII

Páxina de códigos 437 –  
Táboa de códigos orixinal de  
IBM PC para o idioma Inglés

Para ver a táboa de  
caracteres que utiliza a  
consola de windows utilizar o  
comando CHCP

Código ASCII de impresión de 8 bits (con caracteres gráficos) (PC)

izda. → dcha. ↓	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	NUL		SP	0	@	P	p	Ç	É	á	À	⌚	L	⌚	α	≡
	0	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224
0001		DC1	!	1	A	Q	a	q	ü	æ	í	⌚	⌚	⌚	β	±
	1	1	17	33	49	65	81	97	113	129	145	161	177	193	209	225
0010		DC2	"	2	B	R	b	r	é	Æ	ó	⌚	⌚	⌚	Γ	≥
	2	2	18	34	50	66	82	98	114	130	146	162	178	194	210	226
0011	♥	DC3	#	3	C	S	c	s	â	ô	ú	⌚	⌚	⌚	Π	≤
	3	3	19	35	51	67	83	99	115	131	147	163	179	195	211	227
0100	♦	DC4	\$	4	D	T	d	t	ã	õ	ñ	⌚	⌚	⌚	Σ	∫
	4	4	20	36	52	68	84	100	116	132	148	164	180	196	212	228
0101	♣	§	%	5	E	U	e	u	à	ò	ñ	⌚	⌚	⌚	σ	J
	5	5	21	37	53	69	85	101	117	133	149	165	181	197	213	229
0110	♠		&	6	F	V	f	v	ä	ü	ä	⌚	⌚	⌚	μ	÷
	6	6	22	38	54	70	86	102	118	134	150	166	182	198	214	230
0111	BEL		'	7	G	W	g	w	ç	ù	ö	⌚	⌚	⌚	τ	=
	7	7	23	39	55	71	87	103	119	135	151	167	183	199	215	231
1000	BS	CAN	(	8	H	X	h	x	ê	ÿ	ï	⌚	⌚	⌚	φ	°
	8	8	24	40	56	72	88	104	120	136	152	168	184	200	216	232
1001	HT	EM	)	9	I	Y	i	y	ë	Ö	ü	⌚	⌚	⌚	θ	.
	9	9	25	41	57	73	89	105	121	137	153	169	185	201	217	233
1010	LF		*	:	J	Z	j	z	è	Û	⌚	⌚	⌚	⌚	Ω	.
	A	10	26	42	58	74	90	106	122	138	154	170	186	202	218	234
1011	VT	ESC	+	;	K	[	k	{	ï	ç	½	⌚	⌚	⌚	δ	√
	B	11	27	43	59	75	91	107	123	139	155	171	187	203	219	235
1100	FF	FS	<	L	\	l	l	l	î	Ê	¾	⌚	⌚	⌚	∞	η
	C	12	28	44	60	76	92	108	124	140	156	172	188	204	220	236
1101	CR		=	M	]	m	l	l	ÿ	ï	⌚	⌚	⌚	⌚	φ	²
	D	13	29	45	61	77	93	109	125	141	157	173	189	205	221	237
1110	SO		>	N	^	n	^	^	Ä	ß	°	⌚	⌚	⌚	ε	■
	E	14	30	46	62	78	94	110	126	142	158	174	190	206	222	238
1111	SI		/	?	O	_	o	^	À	f	°	⌚	⌚	⌚	Π	SP
	F	15	31	47	63	79	95	111	127	143	159	175	191	207	223	239